Naval Research Laboratory
Washington, DC 20375-5320

NRL/MR/5740--00-8485

Coordinate Transformation for Phased Array Antenna Beam Steering Using GPS and Ship's Motion Data

DIETER R. LOHRMANN

Surface EW Systems Branch
Tactical Electronic Warfare Division

September 18, 2000

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave Blank*)

2. REPORT DATE

September 18, 2000

3. REPORT TYPE AND DATES COVERED

Memorandum Report

4. TITLE AND SUBTITLE

Coordinate Transformation for Phased Array Antenna Beam Steering Using GPS and Ship's Motion Data

5. FUNDING NUMBERS

6. AUTHOR(S)

Dieter R. Lohrmann

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
Washington, DC 20375-5320

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/MR/5740--00-8485

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Naval Sea Systems Command
Washington, DC 20363

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

Beam steering angles of a shipboard phased array antenna are calculated, given the position of the ship, the position and attitude of the antenna on the ship, and the ship's bearing, roll, and pitch angles. The coordinate transformations were written in MATLAB; the pertinent software is added in the Appendix. An example of an application is given.

14. SUBJECT TERMS

Beam steering	Phased array antenna coordinate transformation
GPS	Shipboard

15. NUMBER OF PAGES

23

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std 239-18
298-102

COORDINATE TRANSFORMATION FOR PHASED ARRAY ANTENNA BEAM STEERING USING GPS AND SHIP'S MOTION DATA

Background

In order to steer the pencil beam of a phased array antenna to a target, the target's position needs to be known. This info can come from a tracking radar or from GPS data of both the target and antenna position. When the antenna is installed on a moving platform such as a ship, then this antenna is sometimes mounted on a stabilized platform. Such a stabilized platform is not required if the antenna can be stabilized electronically.

In the following, the algorithms required to steer the pencil beam of a phased array antenna towards a moving target are given. The antenna is mounted on a ship. Input are GPS data of the position of the antenna, the GPS data of the target, and roll, pitch and bearing data of the ship. The programs given were written in MATLAB and can be run on any PC..

Conversion of the geocentric GPS data to longitude, latitude and height

The GPS data can be given in terms of longitude, latitude, and height above sea level, or in Geocentric Coordinates. If the data are given in Geocentric, then these can be transformed to longitude, latitude and height above sea level as follows:
Figure 1 shows the convention for the geocentric convention.
The coordinate system is of the Right Hand Cartesian type with the apex in the center of the earth, the x-axis passing thru the intersect point of the equator and the Greenwich meridian, and the z-axis passing thru the north pole.
Longitude LO, latitude LA, and the height H above the sea surface at the equator are calculated from the Geocentric x y z coordinates :

$$\sin(LA) = z / r \quad , \quad \text{with } r = \text{sqrt}(x^2 + y^2 + z^2)$$

$$tg(LO) = y / x$$

$$H = r - 4E7 / (2*\pi) \quad , \quad \text{with r and H in meters .}$$

This transformation is programmed in the Matlab program gztolola.m (see appendix), with x y z the inputs and LA, LO and H the outputs.

Rotation of the coordinate system

The coordinate system with unity vectors e1 , e2 , e3 in the x,y, and z direction respectively, is rotated resulting in a new coordinate system with unity vectors ep1 , ep2, and ep3 .
Then, [1]

$$ep1 = a11*e1 + a12*e2 + a13*e3$$

$$ep2 = a21*e1 + a22*e2 + a23*e3$$

$$ep3 = a31*e1 + a32*e2 + a33*e3$$

Manuscript approved July 24, 2000.

where

$a11 = \cos(ep1,e1)$ $a12 = \cos(ep1,e2)$ $a13 = \cos(ep1,e3)$

$a12 = \cos(ep2,e1)$ $a22 = \cos(ep2,e2)$ $a23 = \cos(ep2,e3)$

$a13 = \cos(ep3,e1)$ $a23 = \cos(ep3,e2)$ $a33 = \cos(ep3,e3)$

The x1 x2 x3 coordinates of the system e1 e2 e3 are transformed into the system ep1 ep2 ep3 to yield the new coordinates xp1 xp2 xp3 in the rotated system:

$$xp1 = a11*x1 + a12*x2 + a13*x3$$

$$xp2 = a21*x1 + a22*x2 + a23*x3$$

$$xp3 = a31*x1 + a32*x2 + a33*x3$$

Rotation around x1 – axis:

As is illustrated in Figure 2, the x1 x2 x3 coordinate system is rotated counterclockwise when visioning along the x1-axis in the negative direction , counting the angle α positive when rotating in the counter clockwise direction. Then, as can be read from Figure 2,

$a11=\cos(ep1,e1)=1$; $a12=\cos(ep1,e2)=0$; $a13=\cos(ep1,e3)=0$;
$a21=\cos(ep2,e1)=0$; $a22=\cos(ep2,e2)=\cos(\alpha)$; $a23=\cos(ep2,e3)=\cos(90-\alpha)=\sin(\alpha)$
$a31=\cos(ep3,e1)=0$; $a32=\cos(ep3,e2)=\cos(90+\alpha) = -\sin(\alpha)$; $a33=\cos(ep3,e3)=\cos(\alpha)$

This relationship can also be expressed in the form of Innerproducts, remembering that the Innerproduct of two unity vectors equals the cosine of the angle between them. Therefore, a11 = cos(angle between ep1 and e1) = inner product of ep1 and e1. The inner product of two vectors x=[x1 x2 x3] and y=[y1 y2 y3] is x1*y1+x2*y2+x3*y3 . We may write this as "inrprod(x,y) = x1*y1+x2*y2+x3*y3 " .
However, ep1 and e1 must be in the same coordinate system. ep1 appears in the original coordinate system as epp1, where (see Figure 2)
epp1 = [1 0 0] epp2 = [0 $\cos(\alpha)$ $\sin(\alpha)$] epp3=[0 $-\sin(\alpha)$ $\cos(\alpha)$]

and
e1=[1 0 0] e2 = [0 1 0] e3=[0 0 1]

then ,

a11 = cos(epp1,e1) = inrprod(epp1,e1) = 1*1 + 0*0 + 0*0 =1

a12 =cos(epp1,e2) = inrprod(epp1,e2)= 1*0 + 0*1 + 0*0 = 0

a13 = cos(epp1,e3) = 1*0 + 0*0 + 0*1 = 0

a21 = cos(epp2,e1) = 0*1 + cos(α) *0 + sin(α) *0 = 0

a22 = cos(epp2,e2) = 0*0 + cos(α)*1 + sin(α) *0 = cos(α) ...

.

.

.

Now, with all the a_{nm} given, the new coordinates xp1 xp2 xp2 can be calculated from x1 x2 x3 .

The Matlab function xp=rlx(x,α) yields xp, where x is the vector [x1 x2 x3] in the original coordinate system and xp is the vector [xp1 xp2 xp3] in the coordinate system after rotation counterclockwise around the x1 axis , counting the angle positive looking along the positive x1-axis towards the origin.

Rotation around x2 – axis

Similarly, for rotation around the x2 – axis the vector epp is found (Figure1 2A):

epp1=[cos(α) 0 -sin(α)] epp2=[0 1 0] epp3=[sin(α) 0 cos(α)]

The input vector x is transformed into the vector xp by rotation around the x2 axis by angle α by the MATLAB function xp = rly(x,α), see appendix.

Rotation around x3 – axis

For rotation around x3-axis, the vector epp is (Figure 2B):

epp1=[cos(α) sin(α) 0] epp2=[-sin(α) cos(α) 0] epp3=[0 0 1]

The vector x is transformed to the vector xp by rotation around the x3 axis by the MATLAB function xp = rlz(x,α) , see appendix.

Example

Assume that a vector x is given in a coordinate system xx,yy,zz with the xx-axis directed towards the east , the yy-axis towards the north and the zz-axis up. To be calculated is the vector xp referenced to a coordinate system centered on a ship, with the

x-axis parallel to the length axis of the ship, positive towards the bow, the y-axis parallel to the deck, positive towards port, and the z-axis positive up.

The coordinate system xx yy in which the vector x is given is rotated around the zz axis and then around the yy axis until the axis of the rotated system falls on the respective axis of the ship's coordinate system.

The transformation is performed in two steps. First, the vector x is transformed into a coordinate system rotated by the angle $\pi/2$ – beara around the z-axis (see Figure 3). Now the projection of the ship's x-axis onto the xx yy plane lines up with the x-axis of the rotated system. This is performed by

```
xp = rlz(x,π/2-beara)
```

The second step is to rotate the rotated system around its y – axis by the angle pitcha, thereby lining up the x- axis of this rotated system with the ship's x-axis. Now all of the axis of the ship's system are lined up with the axis of the system after the second rotation, and therefore, the resulting vector xp is in reference to the ship's system. The commands are:

```
x=xp
xp = rly(x,pitcha)
```

Be it is assumed that the ship's bearing is 45 degrees, that is, alf = 90 – 45 degrrees. The pitch of the ship be 45 degrees also. Then a vector x=[1 1 sqrt(2)] , when transformed to the ship's coordinate system, should be in the length axis of the ship and have a length of 2, i.e. we should find that xp = [2 0 0].
First transformation: roll around z = axis:
x=[1 1 sqrt(2)];
xp = rlz(x,π/4);
Second transformation: roll around y – axis to compensate pitch:
x = xp;
xp= rly(x, -π/4)

results in

xp=[2 0 0] , which is correct.

Note that the negative of the pitch angle had to be used, because rolling around the y-axis, looking towards the origin, the original coordinate system has to be turned clockwise towards the pitch angle of the ship. However, the convention was that the angle is counted positive for counter clockwise (ccw) rotation; hence, the angle is negative for clockwise (cw) rotation.

Transformation of the Roll angle of the ship

The roll angle "rolla" of the ship is defined as positive, if the roll around the length axis of the ship as seen in direction to the bow is cw. Therefore, with this convention, when seen towards the origin, the rotation is ccw. The transformation is then

```
   x= xp
xp = rlx(x,rolla)
```

Transformation due to antenna position

So far, the antenna was assumed to be located such that its boresite was parallel to the ship's length axis, radiating forward. Let "antposaz" be the angle by which the antenna is rotated around an axis which is perpendicular to the ship's deck, antposaz being positive for cw rotation as seen from on top. Since the rotation is cw, the angle antposaz must be entered negative into rlz. Therefore, the commands to take care of the position of the antenna are

```
 x = xp;
xp = rlz(x,-antposaz)
```

Transformation due to tilt of the antenna boresite

The tilt angle "antposel" of the antenna boresite is defined to be positive when tilted upwards. The coordinates in reference to the antenna are shown in Figure 5. Upward tilt means cw rotation around the y-axis , looking towards the origin. Since the roll algorithm is defined as positive for ccw rotation, the angle antposel must be entered negative into the function rly. Therefore,

```
  x= xp;
xp = rly(x,-antposel)
```

Now we have the original vector x given in coordinates referenced to the antenna, with the center of the coordinate system in the center of the antenna and with the x-axis lined up with the boresite and pointing in the direction of the radiation. The y-axis is pointing towards the left, as seen from on top, looking down on the antenna.
With xp = [xp(1) xp(2) xp(3)] ,

The azimuth steering angle "azim" of the beam is then

azim = atan2(xp(2),xp(1))

It is positive for steering the beam to the left, as seen in direction of propagation.
The elevation angle "elev" of the beam is

elev= atan2(xp(3),sqrt($xp(1)^2$ + $xp(2)^2$))
It is positive for the beam pointing up.

Example:

	Latitude (degrees)	Longitude (degrees)	Heigth (m)
Target:	gpsplati = 29	gpsplong = -85.3	gpsph=150
Ship	gpsslati=29	gpsslong=-85.5	gpssh=5

Bearing of ship: beara= 190 degrees

Roll anngle of ship rolla=-5 degrees

Pitchangle of ship pitcha=5 degrees

Azimuth position of antenna on ship antposaz=270 degrees (port side)

Tilt of antenna boresite is 15 degrees up: antposel=15 degrees

Figure 6 shows the geometry. Both target and ship are at the same latitude; the target is east of the ship by 0.2 degrees, i.e., 12 minutes of arc, or 12 nautical miles, or 22.2 km. At that distance and the target height of 150m, the target is practically on the horizon, as seen from the ship. The data above were entered into testd.m (see appendix). The commands

```
testd
laurant
```

result in

azim=10.0 degrees
elev=-8.6 degrees

for the beam steering angles of the antenna.

Summary of symbols and conventions

x = The vector pointing from the ship antenna to the target. Its length is equal to the distance from the antenna to the target. Its coordinate system is Cartesian, with the x-axis pointing to the east, the y-axis to the north and the z-axis up. This vector is
transformed by several rotational transformations to the coordinate system centered at the antenna.

beara = the ship's bearing angle, positive going cw starting at true north

pitcha = the ship's pitch angle, positive with the bow up

rolla = the ship's roll angle, positive rolling cw as seen looking towards the bow

antposaz = azimuth position of the antenna on the ship, cw starting from the bow. antposaz=0 for the boresite parallel to the length axis, radiation in forward direction

antposel = elevation angle of boresite, referenced to the deck of the ship, positive up.

azim = azimuth pointing angle of beam, offset from boresite in azimuth. Pointing to the left, as seen indirection of propagation renders positive angles.

elev = elevation pointing angle of beam, positive up.

[1] Hütte, Des Ingenieurs Taschenbuch , 28 th edition, p. 115

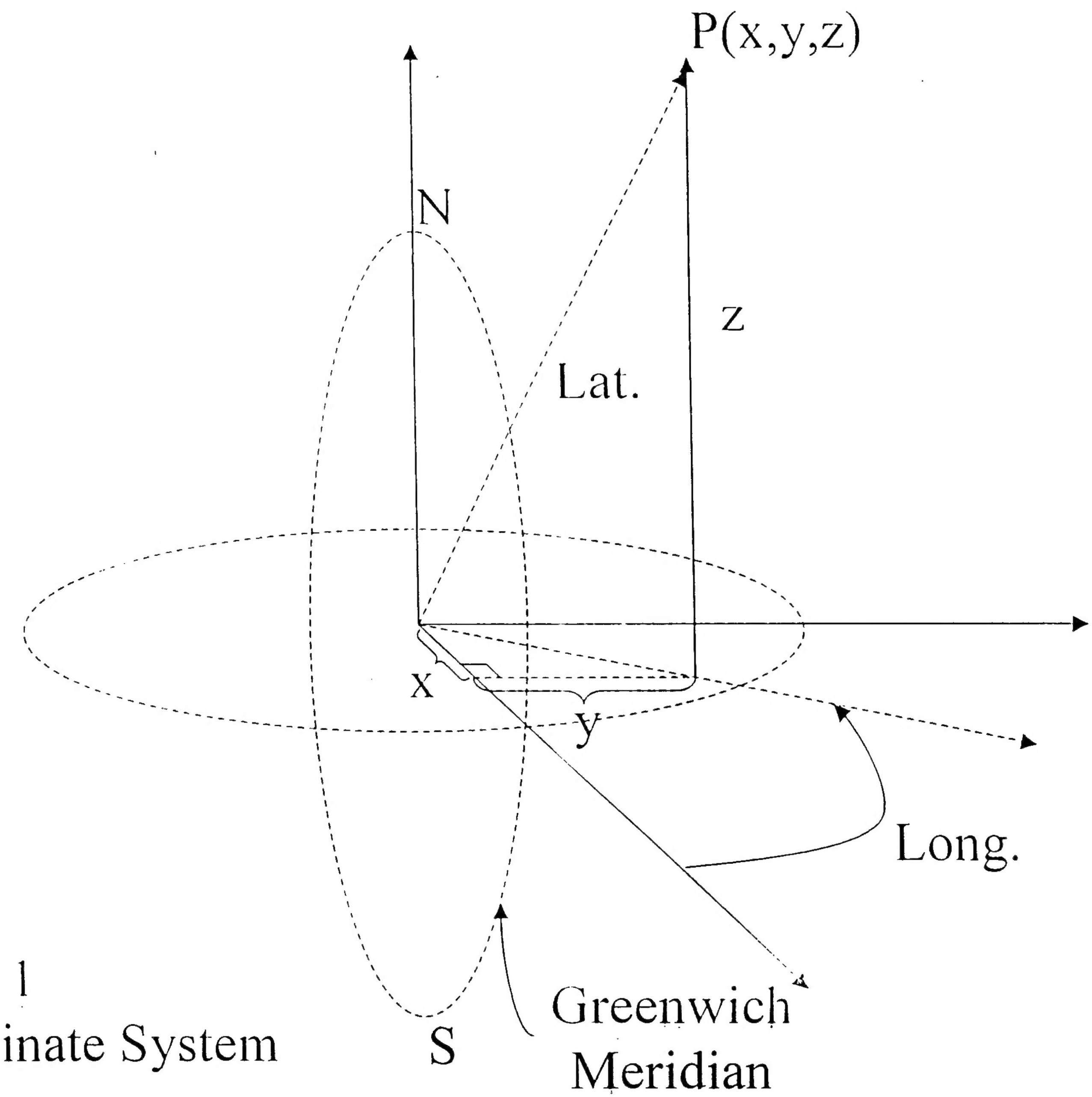
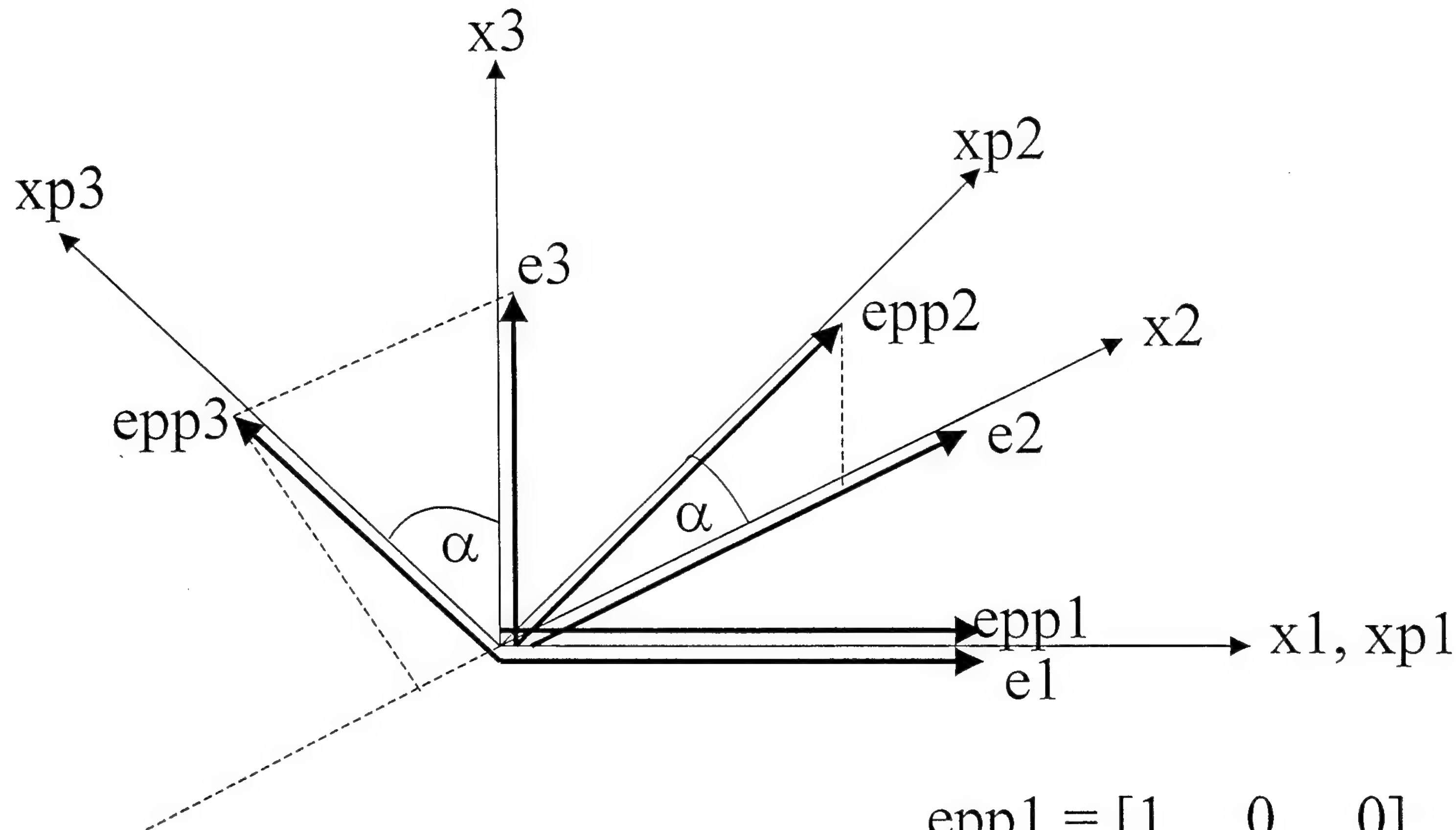
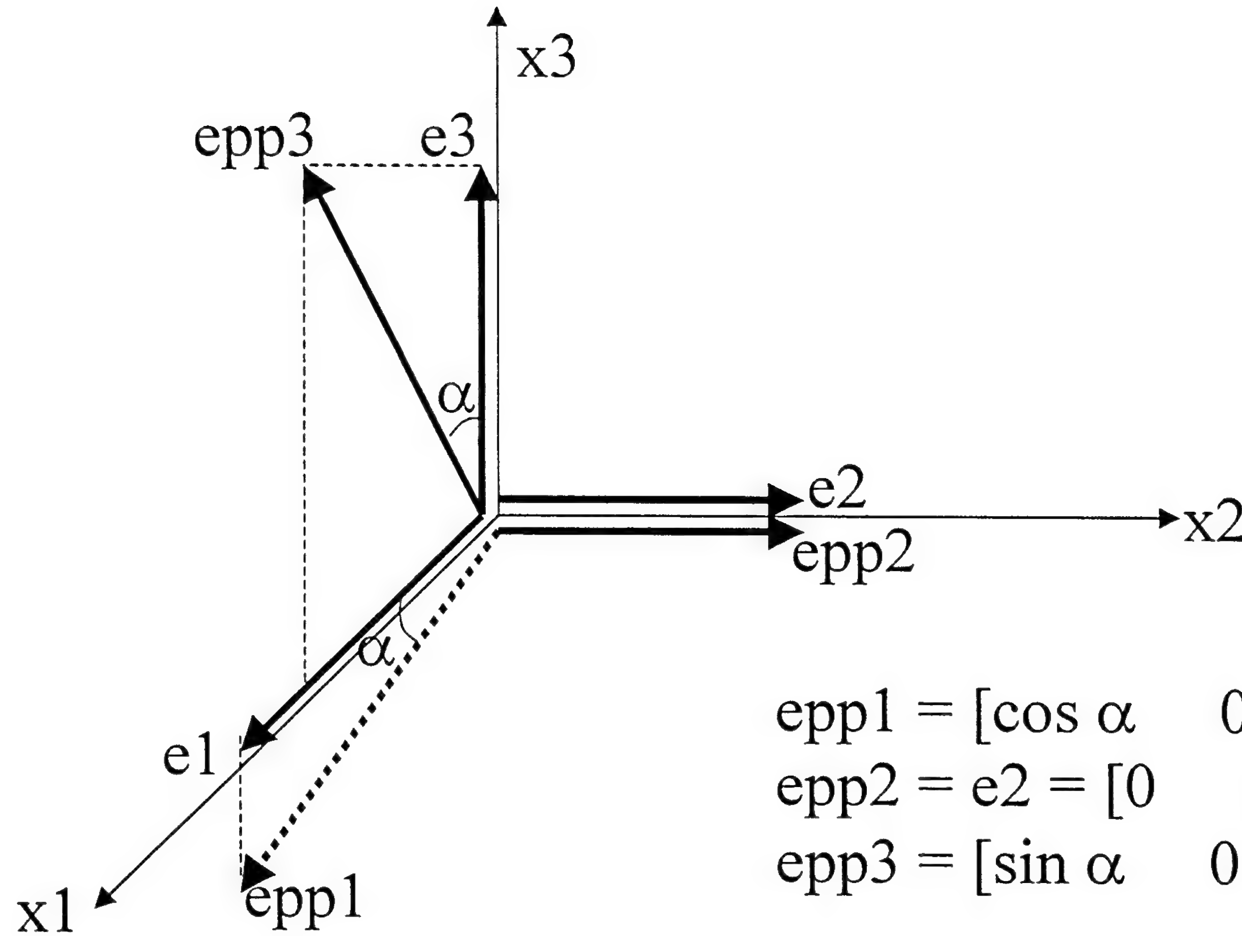


Figure 1
Geocentric Coordinate System



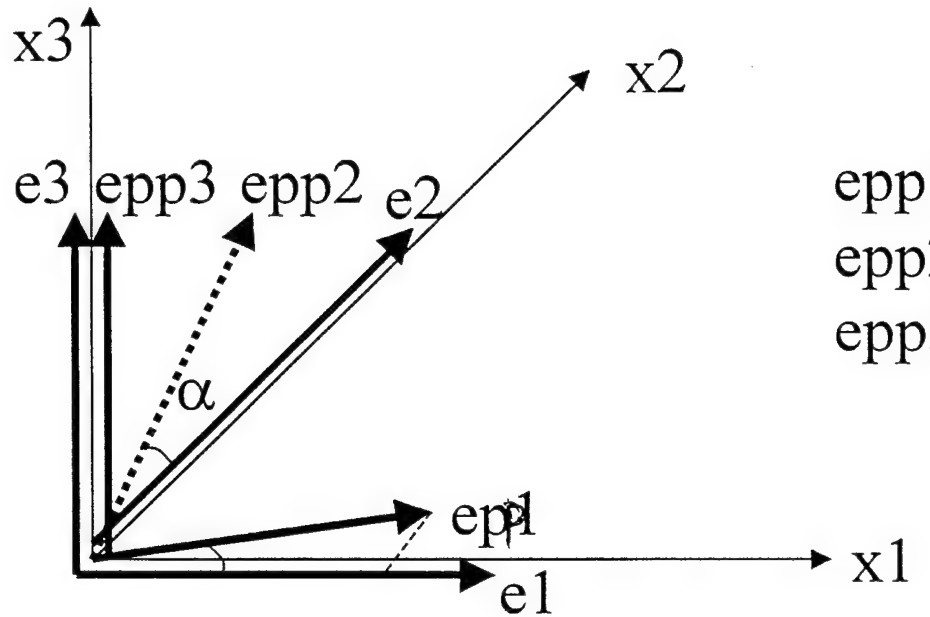
$$
\begin{aligned}
epp1 &= [1 \quad 0 \quad 0] \\
epp2 &= [0 \quad \cos\alpha \quad \sin\alpha] \\
epp3 &= [0 \quad -\sin\alpha \quad \cos\alpha]
\end{aligned}
$$

Figure 2
Rotation Around X1 Axis



$$\begin{aligned}
epp1 &= [\cos\alpha \quad 0 \quad -\sin\alpha] \\
epp2 = e2 &= [0 \quad 1 \quad 0] \\
epp3 &= [\sin\alpha \quad 0 \quad \cos\alpha]
\end{aligned}$$

Figure 2A
Rotation Around X2 Axis



$$
\begin{aligned}
epp1 &= [\cos\alpha \quad \sin\alpha \quad 0] \\
epp2 &= [-\sin\alpha \quad \cos\alpha \quad 0] \\
epp3 &= e3 = [0 \quad 0 \quad 1]
\end{aligned}
$$

Figure 2B
Rotation Around X3 Axis

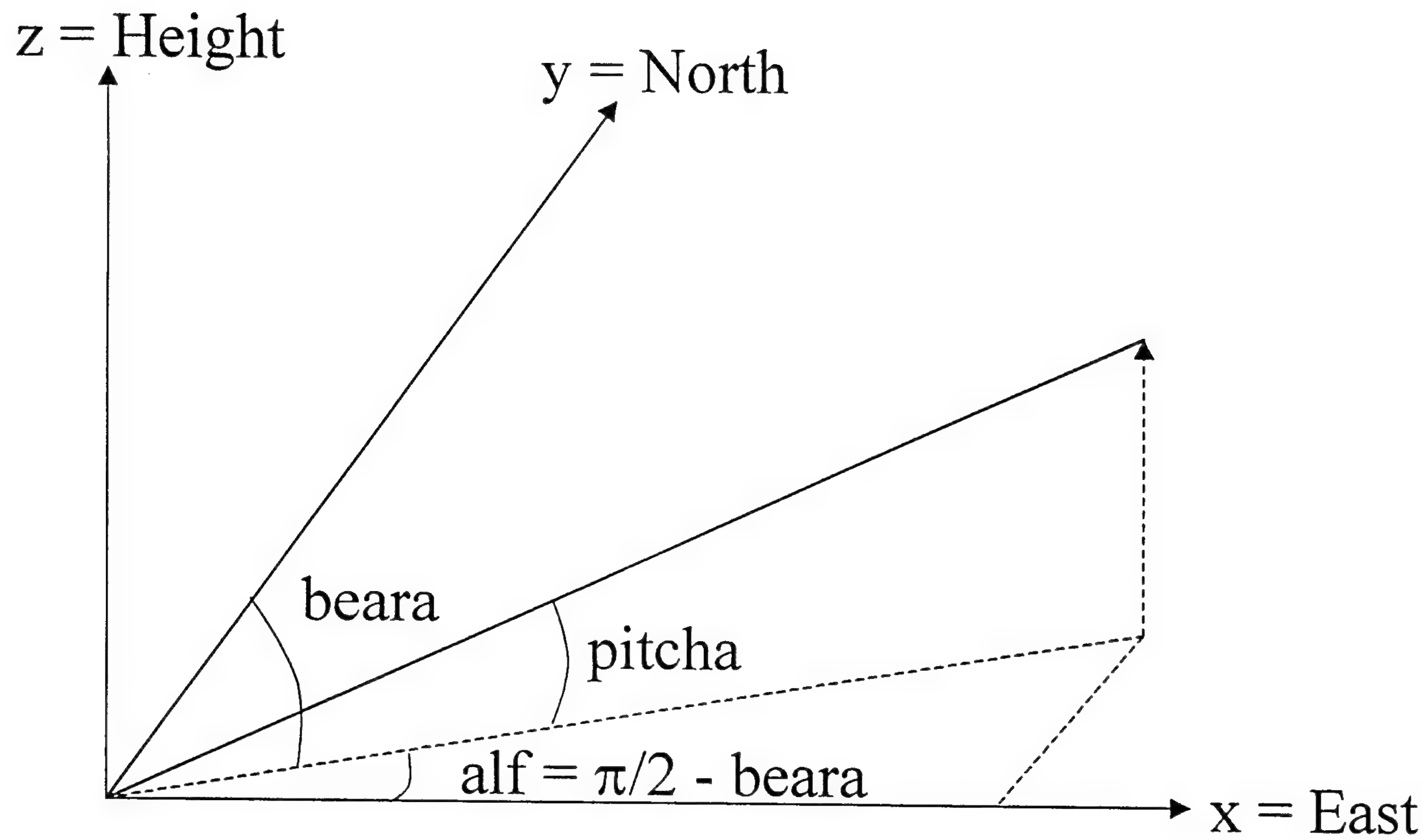


Figure 3

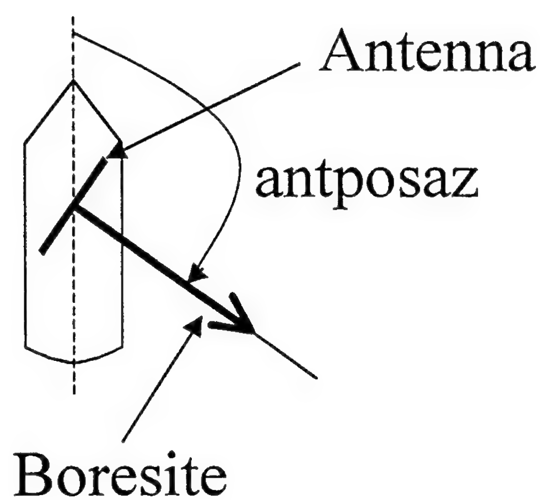


Figure 4
Position of Antenna on Ship

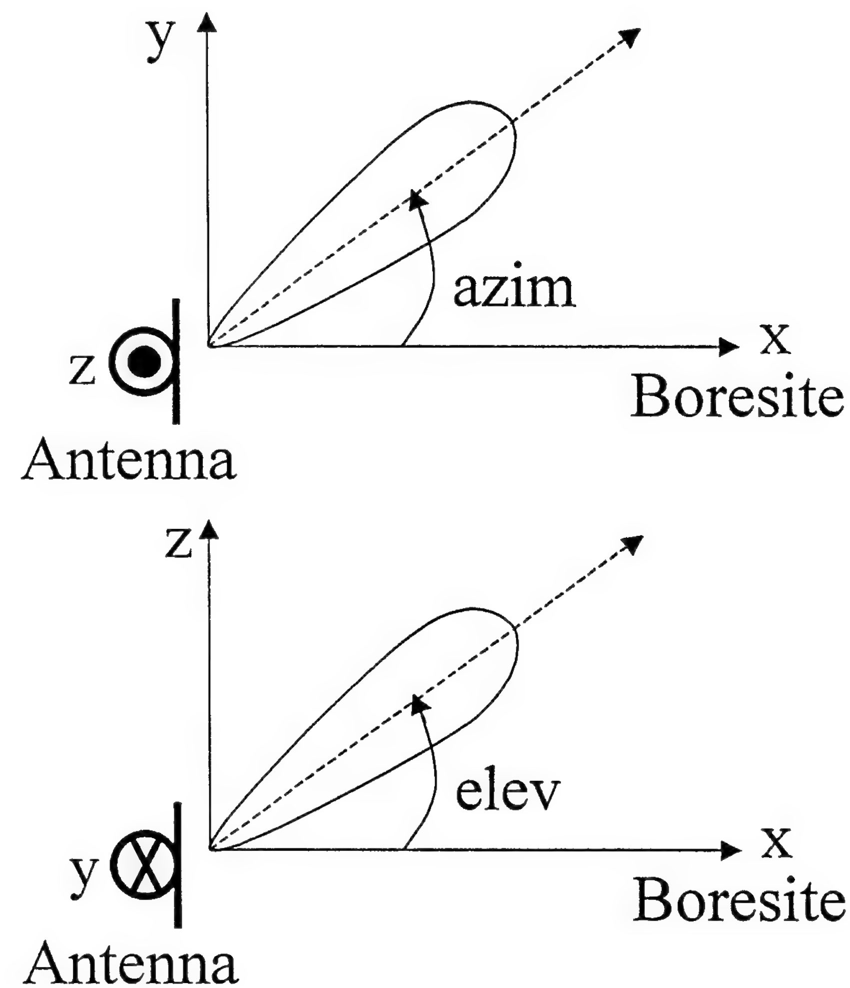


Figure 5
Convention of Antenna Beam Angles

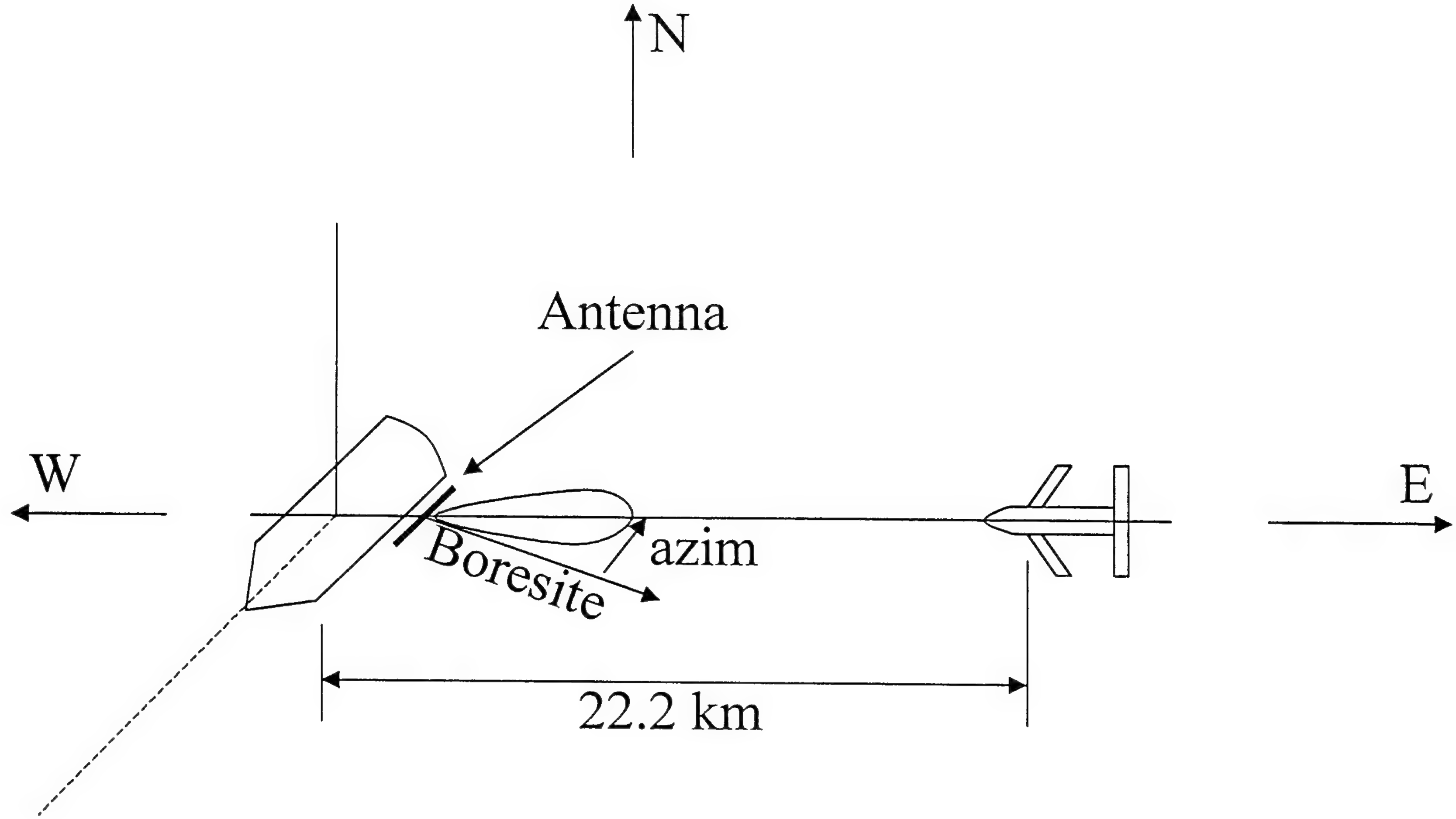


Figure 6
Geometry of Example

APPENDIX

```
% Program name laurant.m
% Dieter R. Lohrmann 04/26/99
% This program calculates azimuth and elevation of antenna
% for test on research vessel Lauren at  NSWC, Panama City Fl
% scheduled for July 15/16 1999
% testdata file in testd.m
% input data:
% input is gps data gpsp of plane and gpss of ship,
% resulting in the directional vector x:
% vector x points from the ship to the aircraft.
% input data:
% gpsplati= gps latitude of plane (degrees)
% gpsplong = gps longitude of plane (degrees)
%            Greenwich meridin is zero, angle
%            is  counted negative in east direction
% gpsslati = gps latitude of ship (degrees)
% gpsslong = gps longitude of ship (degrees)
%            angle increases from zero at G.
%            towards east
%  gpsph    = gps height of plane (m)
% gpssh    = gps height of ship (m)
% beara =  bearing angle of ship vs north (degrees)
% pitcha  =pitch angle of ship  (degrees)
% rolla   =roll angle of ship around its length axis,
%           positive cw looking towards bow of ship (degrees)

% output data:
% azim     = azimuth angle of the antenna beam in reference
%             to the boresite of the antenna array (degrees)
% elev     = elevation angle of the antenna beam in reference to
%              the boresite of the array (degrees).

% intrinsic data:
% antposaz    =  the azimuth position of the antenna on the ship
% antposel    = elevation of the boresite of the antenna array
% The antenna is mounted on the ship's hull; it is not
% on a stabilized platform.
% The vector x points from the ship to the aircraft.
% It is calculated from the gps data as follows:
ax=4e7*cos((gpsslati+gpsplati)*pi/360)*(gpsplong-gpsslong)/360;
ay=(gpsplati-gpsslati)*4e7/360;
% take into account curvature of earth:
az=(gpsph-gpssh)-(ax^2+ay^2)*pi/4e7;


x=[ax ay az];
range=sqrt(ax^2+ay^2+az^2);
% transforming x to the bearing of the ship:
xp=rlz(x,pi/2-beara*pi/180);
% transform vector x to pitch angle:
x=xp;
xp=rly(x,-pitcha*pi/180);
%  Tranformation to roll angle:
x=xp;
xp=rlx(x,rolla*pi/180);
% transformation of x to antenna position:
x=xp;
xp=rlz(x,-antposaz*pi/180);
% transformation of x to tilt of the antenna:
x=xp;
xp=rly(x,-antposel*pi/180);
% This is now the vector of the antenna beam.
% The azimuth angle of the beam from boresite is now:
azim = atan2(xp(2),xp(1))*180/pi
% The elevation angle of the beam:
elev = atan2(xp(3),sqrt(xp(1)^2+xp(2)^2))*180/pi
```

```
% testd.m
gpsplati=29;
gpsplong=-85.3;
gpsslati=29;
gpsslong=-85.5;
gpsph=150.;
gpssh=5.;
beara=190;
pitcha=5;
rolla=-5;
antposaz=270;
antposel=15;
```

```
%  Program name rlx.m
%  Coordinate transformation of threedimensional kartesian
%  Coordinate system. It is rotated around x- axis ccw looking
%  towards origin by angle alf.
%  The unity vecdtors of the original system are e1,e2,e3 .
%  The unity vectors of the transformed system are ep1,ep2,ep3.
%  The input row vector to be transformed into the ep system is x,
%  the transformed row vector is xp  in the p coordinate system.
function xp=rlx(x,alf)
e1=[1 0 0];
e2=[0 1 0];
e3=[0 0 1];
ep1=e1;
ep2=[0     cos(alf)     sin(alf)];
ep3=[0    -sin(alf)     cos(alf)];
a11=sum(ep1.*e1);
a12=sum(ep1.*e2);
a13=sum(ep1.*e3);

a21=sum(ep2.*e1);
a22=sum(ep2.*e2);
a23=sum(ep2.*e3);

a31=sum(ep3.*e1);
a32=sum(ep3.*e2);
a33=sum(ep3.*e3);

aa=[a11 a12 a13
    a21 a22 a23
    a31 a32 a33];

 xp = (aa*x')';
```

```
%  Program name rly.m
%  Coordinate transformation of threedimensional kartesian
%  Coordinate system. It is rotated ccw around y- axis looking
%  towards origin, by angle alf.
%  The unity vecdtors of the original system are e1,e2,e3 .
%  The unity vectors of the transformed system are ep1,ep2,ep3.
%  The input row vector to be transformed into the ep system is x,
%  the transformed row vector is xp  in the p coordinate system.
function xp=rly(x,alf)
e1=[1 0 0];
e2=[0 1 0];
e3=[0 0 1];
ep1=[cos(alf)       0      -sin(alf)];
ep2=e2;
ep3=[sin(alf)       0       cos(alf)];
a11=sum(ep1.*e1);
a12=sum(ep1.*e2);
a13=sum(ep1.*e3);

a21=sum(ep2.*e1);
a22=sum(ep2.*e2);
a23=sum(ep2.*e3);

a31=sum(ep3.*e1);
a32=sum(ep3.*e2);
a33=sum(ep3.*e3);

aa=[a11 a12 a13
    a21 a22 a23
    a31 a32 a33];

 xp = (aa*x')';
```

```
% Program name rlz.m
% Coordinate transformation of threedimensional kartesian
% Coordinate system. It is rotated around z- axis ccw looking
% towards origin, by angle alf.
% The unity vecdtors of the original system are e1,e2,e3 .
% The unity vectors of the transformed system are ep1,ep2,ep3.
% The input row vector to be transformed into the ep system is x,
% the transformed row vector is xp  in the p coordinate system.
function xp=rlz(x,alf)
e1=[1 0 0];
e2=[0 1 0];
e3=[0 0 1];
ep1=[cos(alf)      sin(alf)      0];
ep2=[-sin(alf)     cos(alf)      0];
ep3=e3;
a11=sum(ep1.*e1);
a12=sum(ep1.*e2);
a13=sum(ep1.*e3);

a21=sum(ep2.*e1);
a22=sum(ep2.*e2);
a23=sum(ep2.*e3);

a31=sum(ep3.*e1);
a32=sum(ep3.*e2);
a33=sum(ep3.*e3);

aa=[a11 a12 a13
    a21 a22 a23
    a31 a32 a33];

 xp = (aa*x')';
```

```
% file name gztolola.m
% calculates longitude and latitude (degrees)
% using geocentric coordinates aax,aay,aaz (m)
% The apex of the cartesian right hand
% coordinate system is in the center of the earth
% The x-axis passes thru the Aequator at the
% greenwich meridian. The Y-axis is in the aequat.
% plane, and the z-axis goes thru the north pole.
% h = heigth above surface of earth (m)
r=sqrt(aax^2+aay^2+aaz^2);
la=180/pi*asin(aaz/r)
lo=180/pi*atan2(aay,aax)
h=r-4e7/2/pi
```

[1] Hütte, Des Ingenieurs Taschenbuch, 28 th edition, p. 115 :

6. Koordinatentransformation. $(O; e_1, e_2, e_3)$ und $(O'; e_1', e_2', e_3')$ seien zwei kartesische Koordinatensysteme. Zusammenhang zwischen ihnen ist festgelegt durch

$$\overrightarrow{O'O} = \mathfrak{a} = a_1' e_1' + a_2' e_2' + a_3' e_3'.$$

$$e_i' = a_{i1} e_1 + a_{i2} e_2 + a_{i3} e_3, \quad e_i = a_{1i} e_1' + a_{2i} e_2' + a_{3i} e_3' \quad (i = 1, 2, 3).$$

$$a_{ik} = e_i' e_k = \cos(e_i', e_k).$$

(e_i', e_k) ist der Winkel zwischen der x_i'-Achse und der x_k-Achse. Es gilt:

$$\sum_{l=1}^{3} a_{il} a_{kl} = \delta_{ik} = \begin{cases} 1 & \text{für } i = k \\ 0 & \text{für } i \neq k \end{cases}$$

P sei ein beliebiger Punkt mit dem R.V. $\mathfrak{r}$ im ersten und den R.V. $\mathfrak{r}'$ im zweiten System:

$$\mathfrak{r} = x_1 e_1 + x_2 e_2 + x_3 e_3, \quad \mathfrak{r}' = x_1' e_1' + x_2' e_2' + x_3' e_3',$$

$$\overrightarrow{OP} = \mathfrak{r}, \quad \overrightarrow{O'P} = \overrightarrow{O'O} + \overrightarrow{OP} = \mathfrak{a} + \mathfrak{r} = \mathfrak{r}'.$$

Die Koordinaten x_i von P im ersten System hängen mit den Koordinaten x_i' von P im zweiten System wie folgt zusammen:

$$x_i' = a_i' + a_{i1} x_1 + a_{i2} x_2 + a_{i3} x_3 \quad (i = 1, 2, 3).$$

Bemerkung: Vektor $\mathfrak{a} = \overrightarrow{O'O}$ entspricht einer Parallelverschiebung des Koordinatensystems. Die Matrix (S. 117) $\mathfrak{A} = \|a_{ik}\|$ gibt eine Drehung des Koordinatensystems an. Es ist $\mathfrak{A}' = \mathfrak{A}^{-1}$. Jede Matrix mit dieser Eigenschaft heißt orthogonal.